Al-Azhar University
Faculty of Engineering (Boys)
Systems and Computer Engineering

جامعة الأزهر
كلية الهندسة (بنين)
قسم هندسة النظم والحاسبات

Graduation Project

Parking Space Detection

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Systems and Computer Engineering

Submitted By

1- **Islam Mohamed Mohamed Soliman**
2- **Mohamed Maghawry Ismail**
3- **Mahmoud Osama Mohamed**
4- **Mahmoud Rezk Saad**
5- **Mahmoud Abdulrahman Badr**

Supervised by

Ass. Prof

Gamal Tharwat

2023

Al-Azhar University

Faculty of Engineering (Boys)

Systems and Computer Engineering

جامعة الأزهر

كلية الهندسة (بنين)

قسم هندسة النظم والحاسبات

Graduation Project

Parking Space Detection

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Systems and Computer Engineering

Submitted By

6- **Islam Mohamed Mohamed Soliman**

7- **Mohamed Maghawry Ismail**

8- **Mahmoud Osama Mohamed**

9- **Mahmoud Rezk Saad**

10- **Mahmoud Abdulrahman Badr**

Supervised by

Ass. Prof

Gamal Tharwat

2023

Al-Azhar University
Faculty of Engineering (Boys)
Systems and Computer Engineering

جامعة الأزهر
كلية الهندسة (بنين)
قسم هندسة النظم والحاسبات

Examiner Committee

Name	Rule	Signature
Prof. Dr.	President	
Prof. Dr.	Supervisor	
Dr.	Member	

Date
7th of July, 2023

ABSTRACT

In Our Project We Focus to :

- Finding empty parking slot in less time with fewer efforts is a difficult task for drivers mostly in metropolitan areas. Traffic congestion may occur due to unavailability of empty parking slots. This paper represents an approach for smart parking system which facilitates drivers with available empty parking slots. The system is based on image processing. Using video surveillance, it captures images and using detection it notifies the vehicle driver whether the parking lot is available or not. This system will make parking of vehicle more efficient and less complex by providing a better solution to the existing system
- We use CNN and OpenCV Library In Python

KEYWORDS -----;----------;----------;

ACKNOWLEDGEMENTS

First and foremost, we thank and praise **Allah** for providing us with the patience, strength, well-being, and skills to complete this thesis at one of the most important stages in our lives.

We would like to thank our supervisor, Ass. Prof: Gamal Tharwat for his support and guidance.

We would also like to acknowledge Al-Azhar University for granting us to be among its respectful students.

TABLE OF CONTENTS

ABSTRACT….....iv

ACKNOWLEDGEMENTS.....v

TABLE OF CONTENTS.....vi

LIST OF TABLES….....ix

LIST OF FIGURES….....x

GLOSSARY AND LIST OF ABBREVIATIONS….....xii

CHAPTER 1 INTRODUCTION….....1

1.1 Overview..... 1

1.2 Problem Statement.....2

1.3 Objectives.....2

1.4 Limitation..... 2

1.5 Scope..... 2

1.6 Why We Use CNN..... 2

1.7 Aims and Objectives..... 2

1.8 Methodology..... 2

CHAPTER 2 LITERATURE REVIEW.....4

2.1 Introduction.....4

2.1.1 Counter-Based.....4

2.1.2 Sensor-Based.....6

2.1.3 Vision-Based.....8

2.1.4 CNN.....8

2.2 YOLO.....10

CHAPTER 3 Methodology….....19

3.1 Introduction.....19

3.1.1 ----------.....19

3.1.2 System Architecture.....19

3.2 Requirements.....20

3.2.1 User requirements.....20

3.2.2 System requirements.....20

3.3 Process and Sequence Diagram.....21

3.3.1 Use Case Diagram.....21

3.3.2 Sequence Diagrams.....21

3.4 Database Design.....23

3.4.1 Database Tables.....23

3.4.2 Relationships.....24

3.4.3 Entity-Relationship Diagram.....24

3.4.4 Database Schema.....25

3.5 MVC Architecture.....25

3.6 Proposed Model Architecture.....26

3.7 APIs Communication.....27

CHAPTER 4 IMPLEMENTATION….....28

4.1 Introduction.....28

4.2 Tools & Technologies.....28

4.3 Dataset and Preprocessing.....28

4.3.1 Cleaning using the pre-trained model.....29

4.4 Methodology 30

4.5 Code.....43

CHAPTER 5 RESULTS AND DISCUSSION.....45

5.1 Experimental Consideration.....45

5.2 Experimental Results.....46s

5.2.1.1 Testing Data.....46

5.3 Results Analysis and Discussion.....49

CHAPTER 6 CONCLUSION AND FUTURE WORK….....50

6.1 Conclusion.....50

6.2 Future Work.....50

REFERENCES….....51

APPENDICES.....52

Timeline/Milestones (Gantt Chart).....52

LIST OF TABLES

2.1 Advantages and disadvantages of Sensor-Based
2.1 Advantages and disadvantages of Vision-Based
2.2
2.3
2.4
5.1
5.2
5.3

LIST OF FIGURES

1.1 Parking Space

1.2 Our Project Methodology

2.1 Counter-Based Process

2.2 Sensor-Based Process

2.3 Ultrasonic Sensor

2.4 CNN architecture

2.5 Parking Detection Vision-Based

2.6 YOLOV3 Architecture

2.7 YOLOV3 Architecture

3.1 Data collection From Park 1

3.2 Data collection From Park 2

3.3 the LabelImg User Interfaces when opening

3.4 Image Labelled using LableImg Tool

3.5 Result Labeling Format

3.6 Classes Of Labeling

3.7 Intersection Over Union

3.8 Screenshot taken during training the YOLOv3 model

3.9 Flowchart representation of Algorithm

GLOSSARY AND LIST OF ABBREVIATIONS

Abbreviation	**Meaning**
CNN	Convolutional Neural Network.
YOLO	You Look Only Once
ML	Machine Learning
XML	Extensible Markup Language
mAP	Mean Average Precision
IOU	Intersection Over Union

CHAPTER 1

INTRODUCTION

1.1 Overview

- The main aim of this project is to reduce the traffic in the parking place. Normally we can see in the multiplexes, cinema halls, large industries, and function halls there is problem they have to go and search which line is empty and which line having place to park the vehicle, for parking then they need workers for parking in correct position it is the money consumed process. So to avoid this problem Car Parking System project is implemented.
- Sometimes, there may be an empty space in the parking lot but the driver who sits in the car cannot know where the empty space is exactly located. The reason behind that may be the free spot is far away from the car, or may be the parking space is not visible because of some other cars or big objects hiding the empty space. In the previous days, or may be at some parking lots now there used to be a person who manages the parking spaces, in fact the person who manages also do not have the information about where the empty space is located, they get to know only the total number of empty spaces so the drivers themselves have to look around parking lot and search for the empty space, meanwhile other driver arrives there are many losses such as time, fuel and also temper. So, for these kinds of issues the researchers have developed some solutions like parking lot recognition techniques by making use of some objects like video cameras and sensors which are used to detect an empty and occupied state of the parking space.

Figure 1.1: Parking Space

1.2 Problem Statement

- As the population grows there is an increase in the private vehicles which caused many issues to the people at the crowded areas such as shopping malls, restaurants and universities parking lots. This results in time waste for the drivers in search for the empty parking space which includes atmospheric pollution and frustration of the drivers. On the other hand, most of the other systems show only the total number of spaces empty or occupied rather than showing the exact location.

.

1.3 Objectives

- Design a model which can efficiently detect the Empty and Occupied status of the parking space in an image and a video by use of YOLOv3 object detection.
- Detect a car parking lot occupancy in an image and in the video whether it is Empty or Occupied by using YOLOv3 detector which is represented by overlapping the output in OpenCV and to collect data which is good and large enough to train the model.

1.4 Limitations

- In the YOLO algorithm, the model predicts and verifies the overall image and video in relation to the training data it verifies in the camera point of view at the time of the test
- Unless a few images of the test environment are included in the training dataset, the YOLO model cannot get a good detection.
- Resolution values may change depending on lighting conditions.

1.5 Scope

- In future the model can be trained by increasing the images in the dataset and making huge dataset in such a way that it can detected at different locations and with different environment.
- The other scope is that the system can be developed in such a way that it could give directions to the drivers to the nearby available empty space by using deep learning and AI methods.

why we use CNN ?

- Convolutional Neural Network.
- This Make Greeting Results More Than Other Technique of CPD(Car Parking Detection)
- this technique work by solve the issues will face the CPD like :-
- parking displacement
- non-unified car sizes
- interobject occlusion
- in this project we will perform a Realtime CPD based on CNN by taking videos from surveillance cameras and detecting the occupancy of park by checking the parking space is empty or occupied.
- we will training the park with YOLOV3 model

1.1 Aims and Objectives

The aim of the project is to develop a Model To Avoid Problem Car crowding In a Huge Parking, Malls, Cinema And Crowded Places

1.2 Methodology

- To training and finding a car in the parking space is done using convolutional neural networks. In the proposed method, this would like to collect the images of cars parked in the outdoor parking lot and create a dataset with all the images so that the neural network can detect the car present in the parking space or not accurately

This section outlines the methodology of implementing our proposed bachelor thesis. as shown in Figure 1.

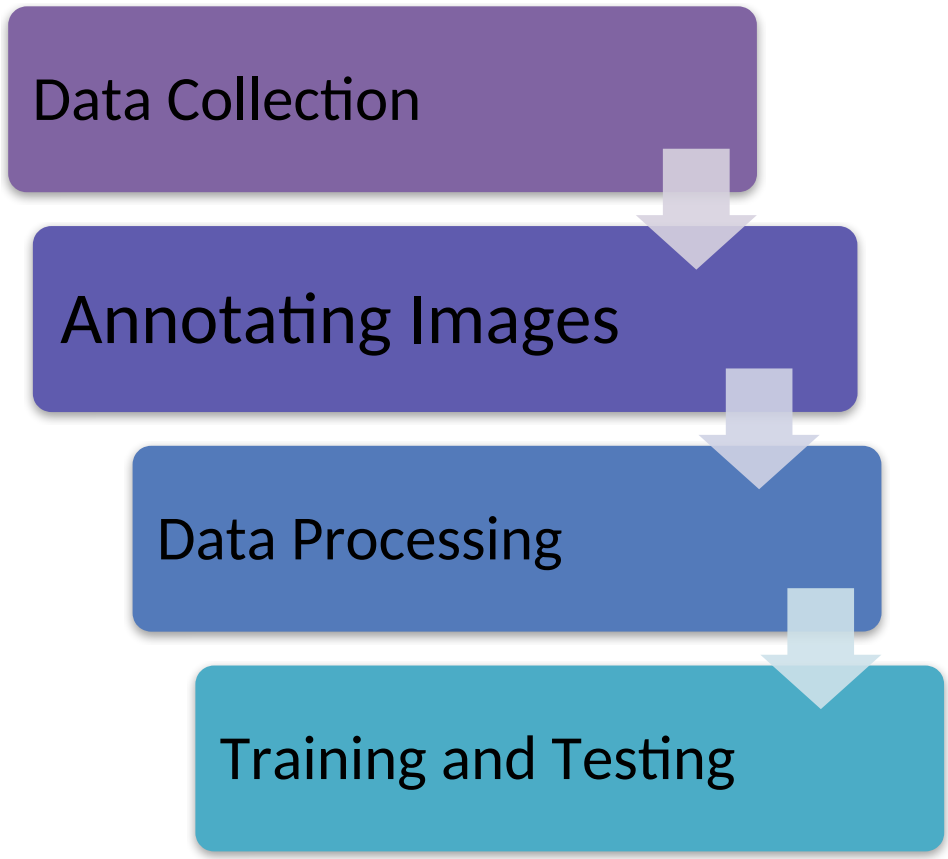

Figure 1.2: Our Project Methodology

CHAPTER 2
LITERATURE REVIEW

2.1 Background

It is a solutions for the parking lot problems are categorized into 3 groups:-

1. Counter-based
2. Sensor-based
3. Vision-based

2.1.1 Counter-Based

Counter-based parking space is a system that uses a counter to keep track of the number of available parking spaces in a given area. The counter can be either electronic or manual, and it is typically placed at the entrance of the parking area.

When a car enters the parking area, the counter is updated to reflect the reduction in available parking spaces. Similarly, when a car exits the parking area, the counter is updated to reflect the increase in available parking spaces. This information is then displayed to drivers looking for a parking space, allowing them to quickly determine whether there are any available spots.

Counter-based parking space systems are typically used in areas where there are a large number of parking spaces, such as parking garages, parking lots, and public parking areas.

They are also commonly used in cities to manage on-street parking spaces.

One of the advantages of counter-based parking space systems is that they can help reduce traffic congestion by allowing drivers to quickly find available parking spaces, rather than circling around looking for a spot. Additionally, they can help reduce air pollution and greenhouse gas emissions by reducing the amount of time that drivers spend looking for parking.

Overall, counter-based parking space systems can help make parking more efficient and convenient for drivers, while also benefiting the environment and reducing traffic congestion.

Disadvantage: we can't use this system on streets and residential parking spaces.

Figure 2.1: Counter-Based Process

2.1.2 Sensor-Based

In the sensor-based systems there are wire and wireless based sensors in which wired systems which mainly depend ultrasonic, infrared light and the wireless based on the magnetic-based sensors which are fixed at the parking lot. Both the wired and wireless are being used mainly at the parking lot like shopping malls which are indoor. Since, these are mainly used at the indoor environments these ground sensors can be used to decide the position of the parking spaces.

So, the solution for this is to use application-focused sensors which can be like magnetic or infrared sensors and pneumatic road tubes. So, these kinds of sensors are very rarely used in some cities. Some paperwork will be required for installing these sensors for the parking space detection and

time will also be wasted and the drivers will face some issues while they are willing to park their cars when the installation of the sensors are going on.

Figure 2.2: Sensor-Based Process

Figure 2.3: Ultrasonic Sensor

Advantages	Disadvantage's
There is high parking efficiency.	It may be a bit confusing for unfamiliar users.
There is no need for driving while looking for an available space.	It requires high installation and maintenance cost.
There are less chances for vehicle vandalism.	It is not recommended for high peak hour volume facilities.

There is a minimal staff requirement if it is used by known parkers.	

Table 2.1: Advantages and disadvantages of Sensor-Based

Related Work

Author: Sangwon Lee, Dukhee Yoon, Amitabha Ghosh

in this paper, they used the magnetic sensors and the ultrasonic sensors for better accuracy and quality recognition of the cars in the parking space.

Author: Yang and Kim (2020)

proposed a system that uses a combination of ultrasonic sensors and a stereo camera to detect parking spaces and obstacles around the vehicle. The system then generates a parking trajectory and controls the vehicle to park in the designated space.

2.1.3 Vision-Based

By using vision-based systems than the sensor-based has two advantages: versatility and lower cost. Rather than using one ground sensor per parking space a smart camera can be used which decreases the cost. Compared to the other systems the accuracy of this system is far better than the others. Therefore, rather than using the other systems it is better to go with the vision-based systems which gives high accuracy results.

Advantages	Disadvantage's

Good for car searching in large parking lots.	Has no real intelligence: follow rules.
Easy for lot occupancy detection, parking space recognition, parking charges collection etc.	Software must be carefully selected and implemented.
Easily adaptable to inspect different products	The lighting conditions must be well designed and implemented.
Maintains consistency of inspection.	
Can be reprogrammed off-line on another machine.	

Table 2.2: Advantages and disadvantages of Vision-Based

CNN

It used to recognize in real-time the occupancy of the parking space which includes the Quality, accurate, efficient and the scalable answer

CNN provides a solution for disturbances like partial occlusions, presence of shadows, variation of

light conditions and exhibits great performance.

In classification it needs computational resources which makes it appropriate to the embedded environment such as Raspberry Pi.

A convolutional neural network is a combination of a large number of hidden layers in which each layer does mathematical computations on the image and gives an output image that is given in input to the network.

the CNN has convolutional layers, which are able to model the spatial correlation of neighboring pixels better than the normal fully connected layers. The classification output is given by the input which is given at the training phase. The model takes lots of training time and also a bit expensive but after the training is done the detection from the network is quite fast and efficient

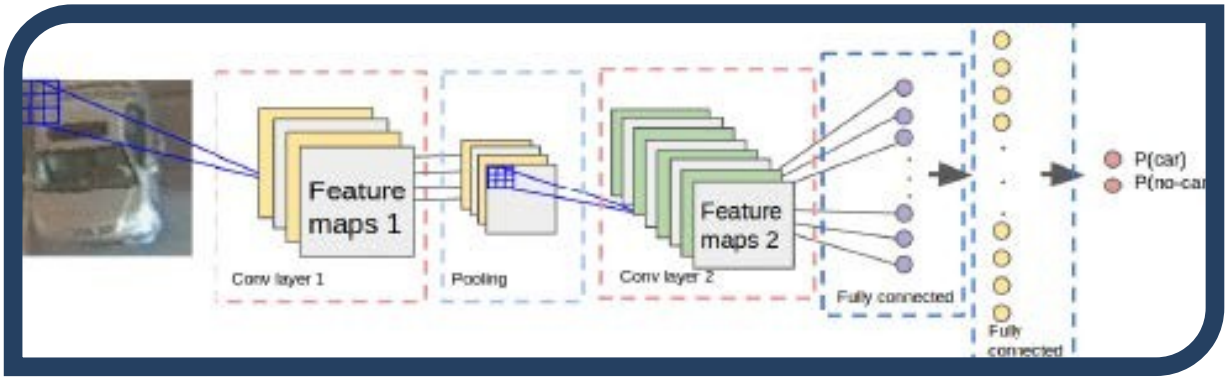

Figure 2.4: CNN architecture

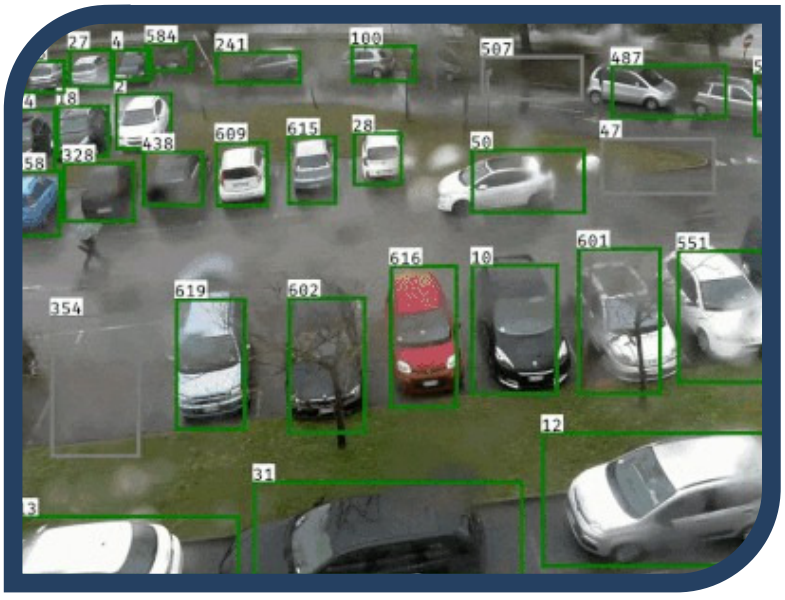

Figure 2.5: Parking Detection Vision-Based

Related Work

Author: G. Amato, F. Carrara, F. Falchi, C. Gennaro and C. Vairo

In this paper, they evaluated the approach for the parking detection, by making using of CNNs to detect the parking lot occupancy by using Raspberry Pi camera.

2.2 YOLO

- You Look Only Once, known as Darknet.
- In the past years, YOLO 9000 used to be the fastest which is also one among the most accurate algorithm. But, before a couple of years it is not the best accurate algorithms when compared with the other algorithms like SSD(Single Shot Multibox Detector) and Retina Net which are the best for accuracy
- Then the next version of the YOLO which is YOLOv3 has been released with some of the changes in the internal architecture
- is one among the faster algorithms for the object recognition.
- YOLO machine learning algorithm is a more accurate version of the original ML algorithm

- YOLO is implemented using the OpenCV deep learning libraries.
- YOLOv3 architecture consists of “53” layered network which is originally trained on ImageNet. Then mainly for the part of detection, “53” more layers are added onto it, and is given by a “106” fully convolutional underlying architecture for YOLOv3
- In the YOLOv3 algorithm the detections can be done at 3 different scales which is considered as the most important feature in the entire algorithm.
- YOLOv3 architecture the detection of the object in the image is done at 3 different sizes and at 3 different layers in the architecture.

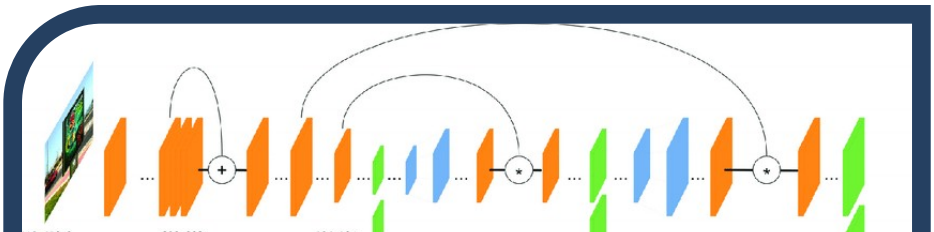

Figure 2.6: YOLOV3 Architecture

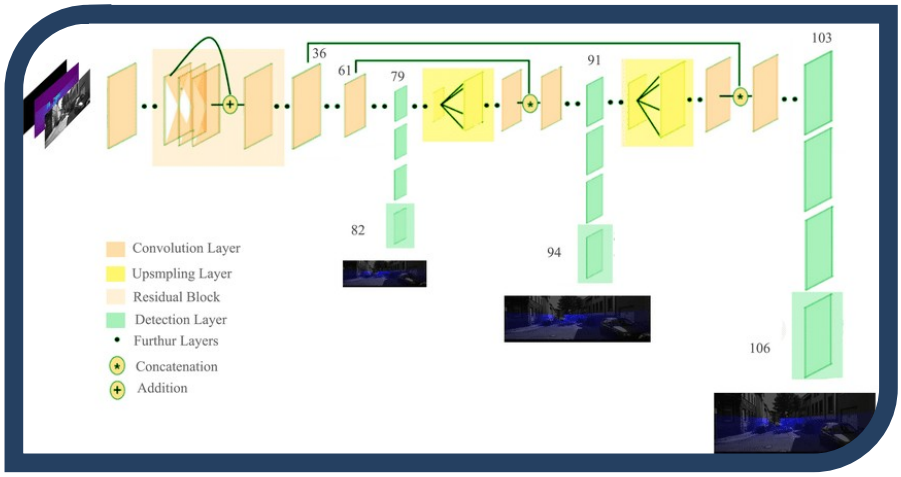

Figure 2.7: YOLOV3 Architecture

It is fully convolutional neural network which gives the output by applying a “1 x 1” kernel on a feature map .
The shape of the detection kernel is given by “1 x 1 x (B x (5+C))”.
B represents the no. of the bounding boxes which are present in a cell on the feature map which can predict.
5 represents the 4 boundary box attributes and 1 for the confidence of the object
C denotes the no. of classes
The YOLOv3 which is trained on COCO dataset
It gives B = 3 and C = 80, so which gives the size of the kernel as
“1 x 1 x (3 x(5+80))” –> “1 x 1 x 255”.

Example

the image which is given as input is of size “416 x 416”,
as mentioned earlier the algorithm makes the predictions at 3 different layers of the network which are done by down sampling the data by 32, 16 and 8.
In the YOLOv3 algorithm detection is done at 3 different layers.

- 1st detection will take place at the 82th layer, by giving a feature map of “13 x 13 x 255”. Stride 32

13 x 13 → 416/32=13

- 2nd detection will take place at the 94th layer, by giving a feature map of “26 x 26 x 255”. Stride 16

26 x 26 → 416/16=26

- 3rd detection will take place at the 106th layer, by giving a feature map of “52 x 52 x 255”. Stride 8

52 x 52 → 416/8=52

- YOLOv3 makes use of 3 anchor boxes,
- YOLOv3 totally makes use of 9 anchor boxes,
- These anchor boxes are arranged one by one it calculate the probability of the object
- YOLOv3 makes predictions 10times the number of boxes done by YOLOv2.
- Each bounding box in a YOLO algorithm consist of four descriptions:
- Centre of the bounding box (b_x, b_y)
- Width (b_w)
- Height (b_h)
- Class of object (c)
- The Probability that there is an object in the bounding box (p_c)

CHAPTER 3

Methodology

We Have 4 Steps To Execution Image:

3.1 Data Collection[1]

- The PKLot dataset contains 12,416 images of parking lots and 695,899 images of parking spaces segmented from them, extracted from surveillance camera frames. There are images on sunny, cloudy, and rainy days and the parking spaces are labeled as occupied or empty. We have converted the original annotations to a variety of standard object detection

formats by enclosing a bounding box around the original dataset's rotated rectangle annotations.

- All images were acquired at the parking lots of the Federal University of Parana (UFPR) and the Pontical Catholic University of Parana (PUCPR), both located in Curitiba, Brazil. The protocol used to construct the PKLot dataset is composed of three steps, as follows:
  1- Image Acquisition
  2- Labeling
  3- Segmentation

Figure 3.1: Data collection From Park 1

Figure 3.2: Data collection From Park 2

3.2 Data Collection

- We annotated images with two different classes which are “Empty” and “Occupied”.
- We Store Data in a Label:
- For labelling the image we used **LabelImg** software which is an open source.
- In this software the images can be annotated easily as it is user friendly.
- Firstly, the software contains some pre-defined classes which should be changed according to our classes, we have named the classes as “Empty” and “Occupied”.
- The software is set to YOLO format and we annotated the images where there is no car in a particular slot as “Empty” and where there is a car as “Occupied”.
- The data of the bounding boxes that we annotated along with the two classes are saved in the txt format which will save the coordinates of the bounding boxes.

LableImg Tool

- **What is Data labeling?**
- One of the most important part in the data science process is data pre-processing which makes data scientists put lots of time and effort into. Particularly, in the context of machine learning, data pre-processing requires a step of labeling.
- This is the step to detect and label data sample into multi-classifications so that the labeled data can be used in further machine learning process.
- Two typical labeling data type in ML projects are text and image annotation.

What is LabelImg?

- **LabelImg** is a graphical image annotation tool which provides images with bounding boxes after labeling. It is written in Python and uses Qt (one of the most common GUI for python) for its graphical interface.
- The output labeled data are saved as XML files in PASCAL VOC format, the format used by ImageNet. Also, YOLO format is supported in this tool.

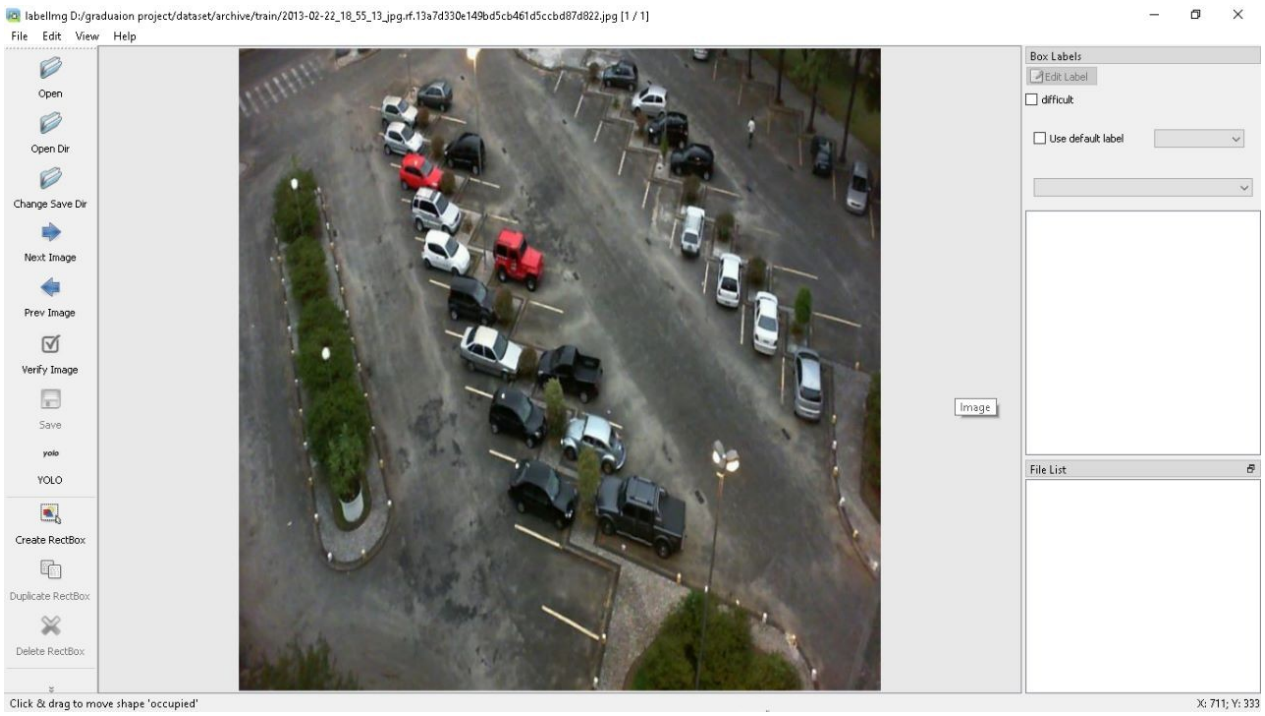

Figure 3.3: the LabelImg User Interfaces when opening

Figure 3.4: Image Labelled using LableImg Tool

Why do we use LabelImg?

- It is open-source then it is free to use.
- It has both online and offline versions. The offline interface is user-

friendly and allows the user to make fast annotations.

- The output annotation images are saved separately as XML files along with Pascal VOC format or YOLO format. These are common formats supported by many ML packages.
- After installation, it is easy to use with simple instructions.
- It can process a large number of images when compared to other free apps such as Cloud Vision API (the free-offline version is limited up to only 2000 image files) or Label box (the free version only allows maximum of 5000 images to be processed).
- It can be run in multiple operating systems such as Linux, Mac or Windows.
- It is written in Python then it can interact with other python packages and is easy to be used in python applications for further development.

- **this tool still has some limitations:**
- Bounding box (rectangle shape) is the only region shape to be used. Therefore, it cannot give an exactly detection for complicated objects with curved lines…
- The installation requires some tricks we need to know. Therefore, I will present how to install this tool step by step in the next session.

3.3 Data Processing

- For YOLOV3 detector, there is a different format for the values to be fed into the model. These values are saved in the txt format which contains some of the specified classes and some values. These values can be as below.
- The order of the values in txt file is class, x, y, w, h.
- “where x = Absolute x / width of total image.
- y = Absolute y / height of total image.
- w = Absolute width / width of the total image.
- h = Absolute height / height of total image.”
- Where Absolute x, Absolute y, Absolute width, Absolute height are given below
- “Absolute x = (Xmin + (Absolute width/2))
- Absolute y = (Y_{min} + (Absolute height/2))
- Absolute width = abs (Xmax - Xmin)
- Absolute height = abs (Y_{max} - Y_{min})”

```
1 0.322656 0.331250 0.092188 0.075000
1 0.703906 0.314063 0.057813 0.100000
1 0.660937 0.249219 0.059375 0.089063
1 0.621094 0.190625 0.064062 0.075000
0 0.582031 0.139063 0.057813 0.078125
1 0.521875 0.064062 0.040625 0.068750
1 0.554688 0.106250 0.040625 0.075000
0 0.364063 0.407813 0.137500 0.084375
0 0.396875 0.492969 0.143750 0.092188
0 0.434375 0.585156 0.150000 0.098437
0 0.458594 0.689063 0.135937 0.115625
0 0.627344 0.729688 0.164062 0.143750
0 0.575000 0.616406 0.096875 0.107813
0 0.535156 0.507812 0.107813 0.090625
1 0.486719 0.424219 0.085938 0.076563
1 0.486719 0.423438 0.114062 0.075000
0 0.457031 0.355469 0.098437 0.082812
1 0.426563 0.286719 0.093750 0.064062
0 0.395313 0.228906 0.100000 0.064062
0 0.378125 0.177344 0.100000 0.060937
1 0.338281 0.126562 0.070312 0.040625
1 0.275781 0.205469 0.089063 0.060937
1 0.767188 0.040625 0.040625 0.062500
1 0.810937 0.096875 0.034375 0.078125
1 0.854688 0.141406 0.040625 0.095312
0 0.905469 0.180469 0.048438 0.095312
0 0.961719 0.221875 0.051562 0.090625
```

Figure 3.5: Result Labeling Format

```
occupied
empty
```

Figure 3.6: Classes Of Labeling

YOLO format of txt files

- Note: the number ‘0’ denotes the class “Empty” and the number ‘1’ denotes the class “Occupied” and the other values denotes the metrics x,y,w,h.

3.4 Training and Testing

- As the deep learning requires hardware with high end specifications for training.
- We have used the pre-trained weights of YOLOV3 which are trained on COCO dataset which consists of 80 classes and images are more than 80000. The training of our model on the dataset was done by the previous steps. The filters of YOLOV3 configuration file should be set according to the below YOLOV3 filters formula
- **"Filters = (classes + 5) * 3"**
- As there are two classes in our case, the classes are set to our number of objects that is two in each of 3 yolo layers, the filters which are in convolutional layer just before the 3 yolo layers are set to 21. The YOLOv3 configuration file consists of batch number which is set to 64, subdivisions to 16, learning rate to 0.001, saturation to 1.5 and exposure to 1.5. The resolution of the input image is resized to a width and height of 608*608. And the training of our model is done, but as YOLOv3 does not limit the number of iterations to be done, the system continuously trains the model. Usually 2000 iterations are sufficient for each class(object), but the iteration number should not be less than 4000 in total. But, for a better precision the training should be stopped for the iteration where the average loss no longer decreases or remains constant. So, when the training should be actually stopped depends upon the average loss function.

-
- The average loss function changes gradually from a higher value to the lower value and remains constant, then the training can be stopped.
- The weights file is generated and saved for every 1000 iterations and this can be changed in the configuration file for our desired number of iterations.
- After the training is done the mAP (Mean Average Precision) is calculated for the validated images.
- This should be done for every weight so that we can know the mAP for each and every weight and the weights file whose mAP is higher can be used for testing the image for good detection.
- Thresh is considered as the minimum threshold of the IOU(Intersection Over Union) during the training. IOU(Intersection Over Union) is determined to be the fraction of area of union between the anchor box and the ground truth.

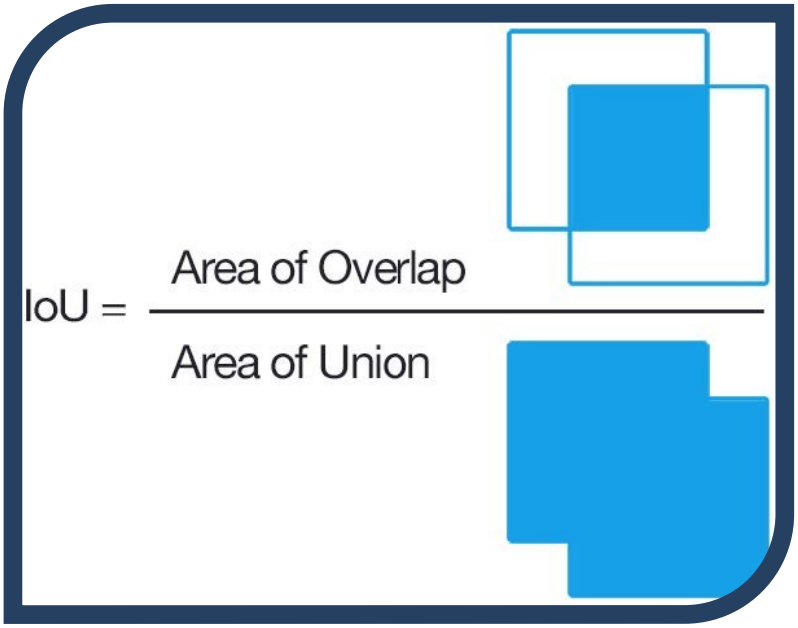

Figure 3.7: Intersection Over Union

```
arvn@isemes: ~/Desktop/darknet-master
Region 106 Avg IOU: 0.347823, Class: 0.460561, Obj: 0.612055, No Obj: 0.521389, .5R: 0.250000, .75R: 0.000000,  count: 8
Region 82 Avg IOU: 0.343092, Class: 0.560348, Obj: 0.574712, No Obj: 0.560946, .5R: 0.111111, .75R: 0.055556,  count: 18
Region 94 Avg IOU: 0.279679, Class: 0.517110, Obj: 0.466742, No Obj: 0.532823, .5R: 0.052632, .75R: 0.000000,  count: 19
Region 106 Avg IOU: 0.144648, Class: 0.493943, Obj: 0.437816, No Obj: 0.521122, .5R: 0.000000, .75R: 0.000000,  count: 1
Region 82 Avg IOU: 0.454006, Class: 0.505768, Obj: 0.585205, No Obj: 0.560749, .5R: 0.363636, .75R: 0.000000,  count: 11
Region 94 Avg IOU: 0.363805, Class: 0.542370, Obj: 0.540078, No Obj: 0.532969, .5R: 0.212121, .75R: 0.000000,  count: 33
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.521478, .5R: -nan, .75R: -nan,  count: 0

 3: 2217.521484, 2217.309326 avg loss, 0.000000 rate, 5.833347 seconds, 192 images
Loaded: 0.000016 seconds
Region 82 Avg IOU: 0.401508, Class: 0.547913, Obj: 0.572229, No Obj: 0.560689, .5R: 0.133333, .75R: 0.000000,  count: 15
Region 94 Avg IOU: 0.299048, Class: 0.573037, Obj: 0.441787, No Obj: 0.533077, .5R: 0.083333, .75R: 0.000000,  count: 36
Region 106 Avg IOU: 0.287174, Class: 0.624580, Obj: 0.720109, No Obj: 0.521554, .5R: 0.000000, .75R: 0.000000,  count: 1
Region 82 Avg IOU: 0.324613, Class: 0.509255, Obj: 0.553081, No Obj: 0.560891, .5R: 0.133333, .75R: 0.000000,  count: 15
Region 94 Avg IOU: 0.384528, Class: 0.570727, Obj: 0.525876, No Obj: 0.533204, .5R: 0.370370, .75R: 0.000000,  count: 27
Region 106 Avg IOU: 0.263588, Class: 0.394126, Obj: 0.542687, No Obj: 0.521282, .5R: 0.000000, .75R: 0.000000,  count: 1
Region 82 Avg IOU: 0.292775, Class: 0.487609, Obj: 0.616293, No Obj: 0.560947, .5R: 0.090909, .75R: 0.000000,  count: 11
Region 94 Avg IOU: 0.350639, Class: 0.565962, Obj: 0.508374, No Obj: 0.533101, .5R: 0.184211, .75R: 0.000000,  count: 38
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.520996, .5R: -nan, .75R: -nan,  count: 0
Region 82 Avg IOU: 0.355499, Class: 0.540567, Obj: 0.530041, No Obj: 0.561170, .5R: 0.181818, .75R: 0.000000,  count: 11
Region 94 Avg IOU: 0.373945, Class: 0.622908, Obj: 0.497447, No Obj: 0.532589, .5R: 0.187500, .75R: 0.062500,  count: 16
Region 106 Avg IOU: 0.360621, Class: 0.521521, Obj: 0.631306, No Obj: 0.520904, .5R: 0.500000, .75R: 0.000000,  count: 4
Region 82 Avg IOU: 0.378471, Class: 0.539884, Obj: 0.507564, No Obj: 0.561241, .5R: 0.333333, .75R: 0.000000,  count: 9
Region 94 Avg IOU: 0.355283, Class: 0.635396, Obj: 0.522794, No Obj: 0.532653, .5R: 0.151515, .75R: 0.000000,  count: 33
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.520930, .5R: -nan, .75R: -nan,  count: 0
Region 82 Avg IOU: 0.266459, Class: 0.465603, Obj: 0.563383, No Obj: 0.561529, .5R: 0.058824, .75R: 0.000000,  count: 17
Region 94 Avg IOU: 0.373580, Class: 0.631968, Obj: 0.483301, No Obj: 0.533053, .5R: 0.185185, .75R: 0.037037,  count: 27
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.521213, .5R: -nan, .75R: -nan,  count: 0
```

3.5 Flow Chart:

The flow chart for training and testing the YOLOv3 model and representing it in OpenCV is as follows in the below

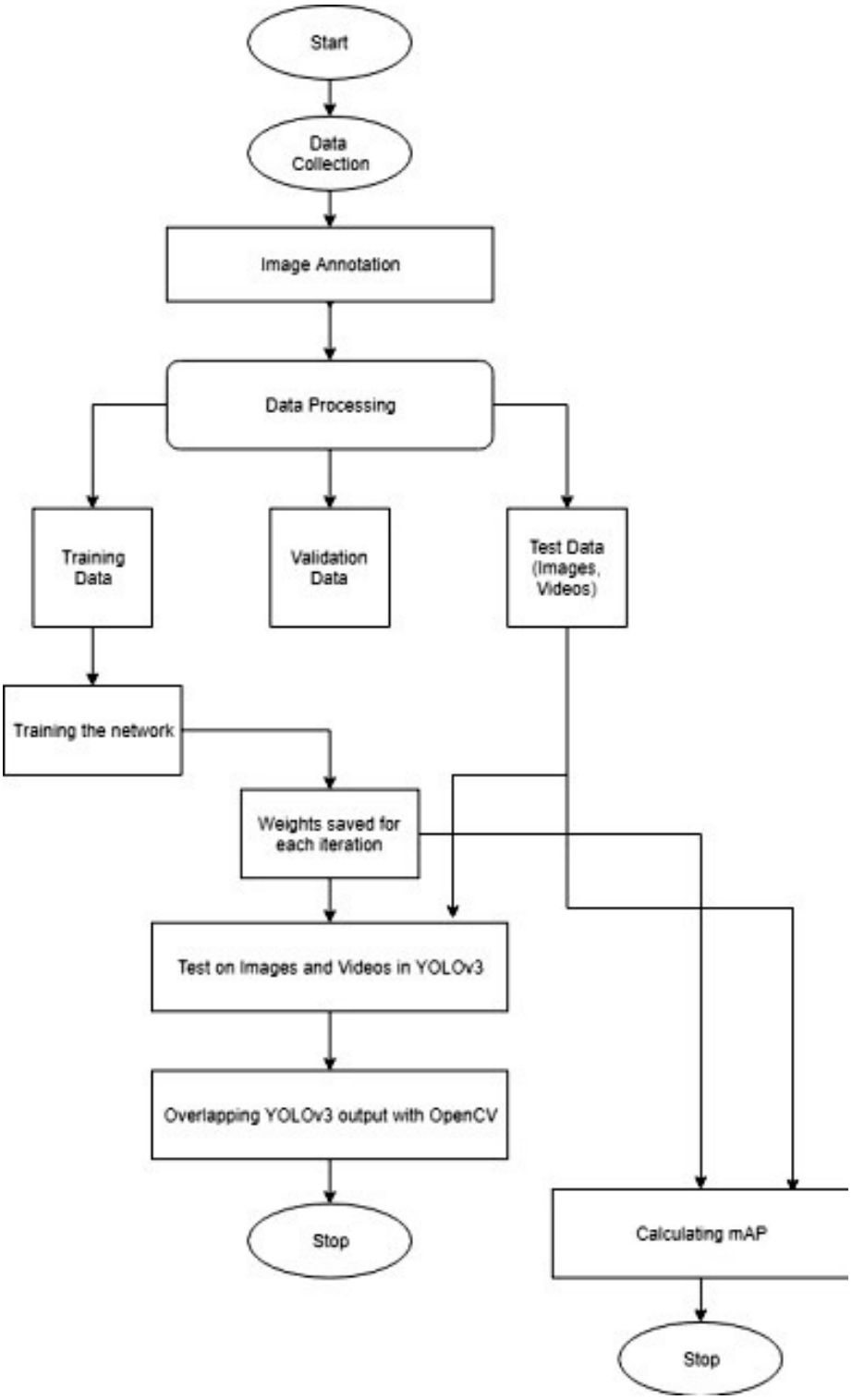

Figure 3.9: Flowchart representation of Algorithm

REFERENCES

1- **[Almeida et al.(2013)Almeida, Oliveira, Silva, Britto Jr & Koerich] Almeida, P., Oliveira, L. S., Silva, E., Britto Jr, A. S., & Koerich, A. (2013). Parking space detection using textural descriptors. In Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on (pp. 36033608). doi:10.1109/SMC.2013.614.**

2-